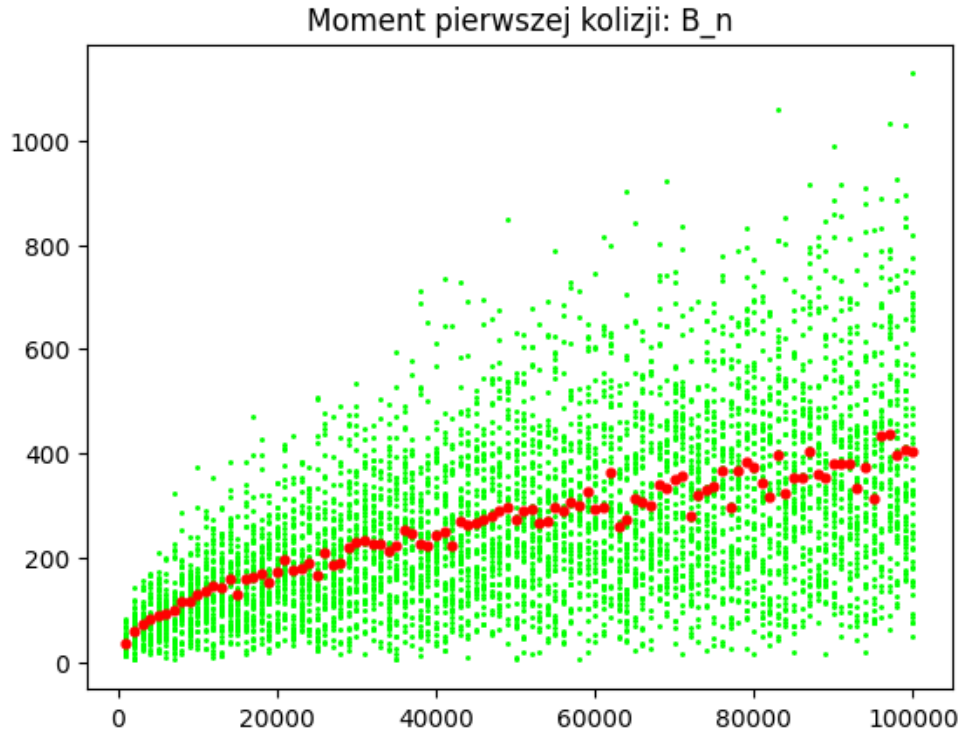
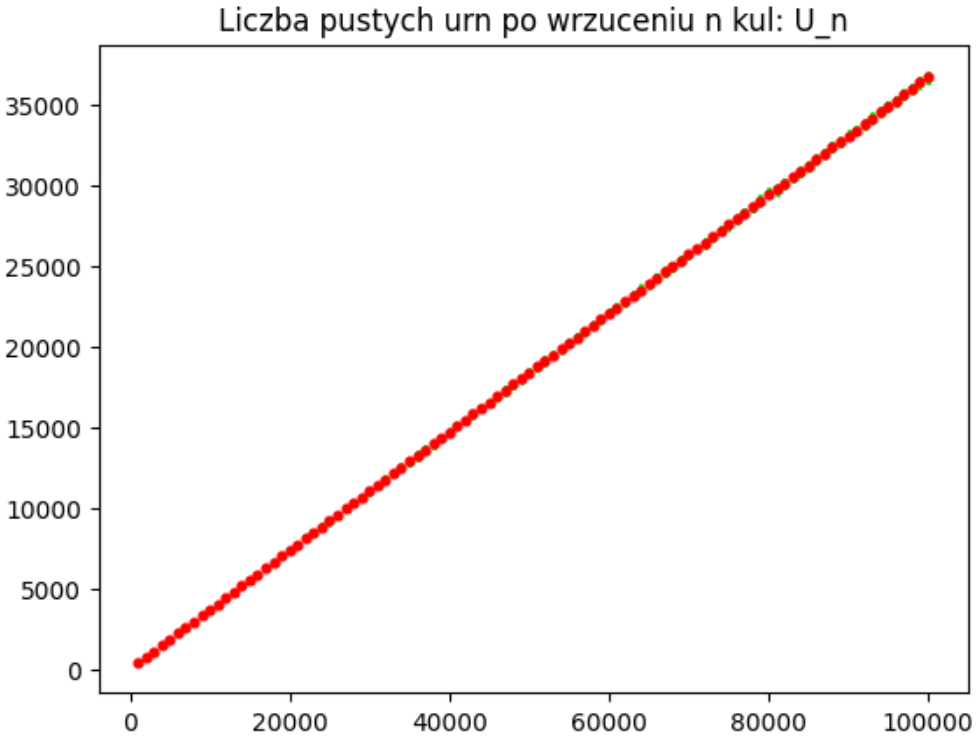
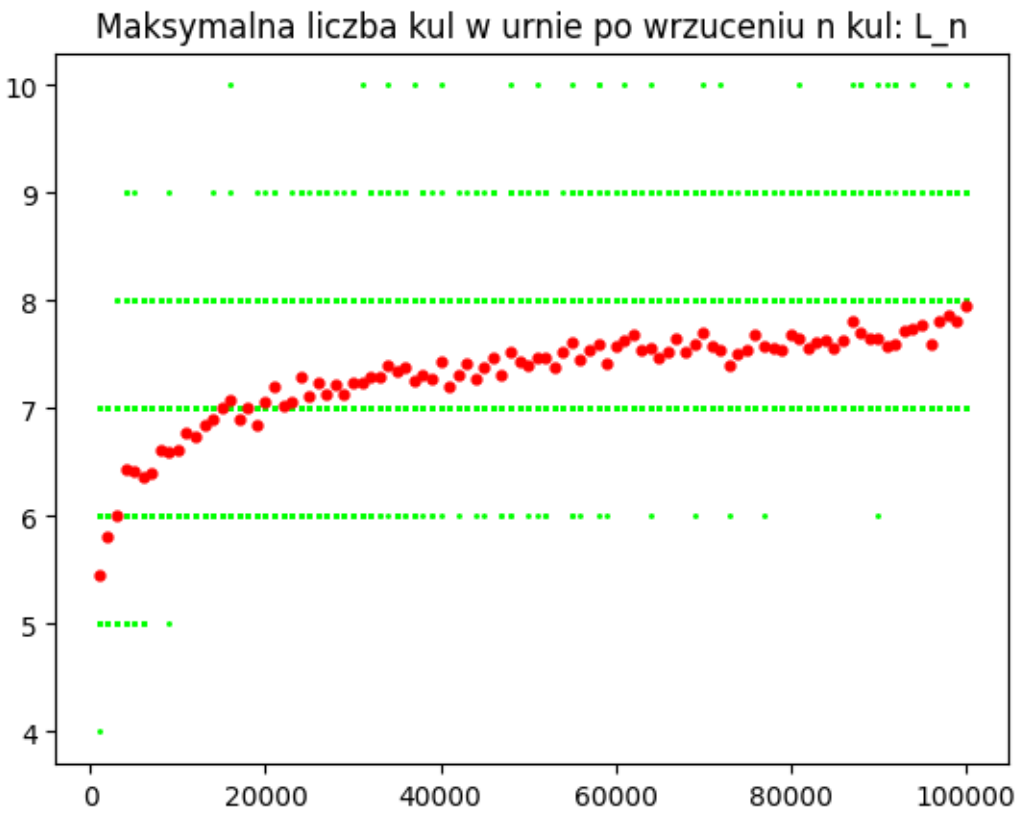
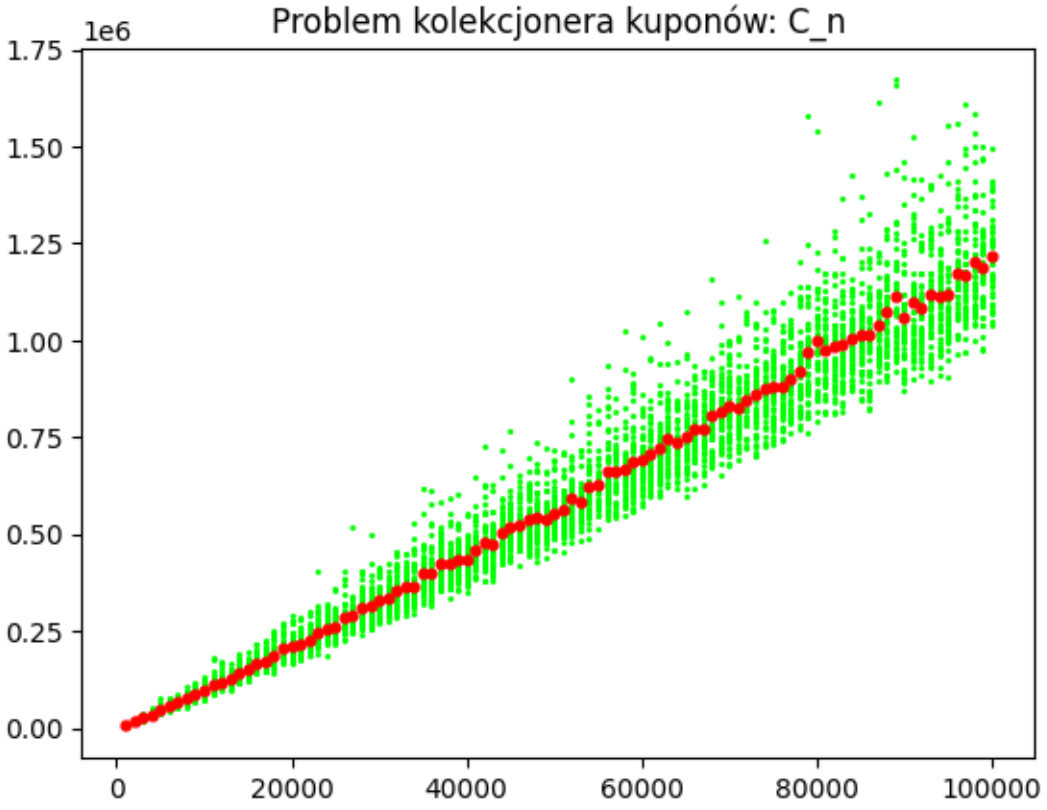
Sprawozdanie do zadania 2

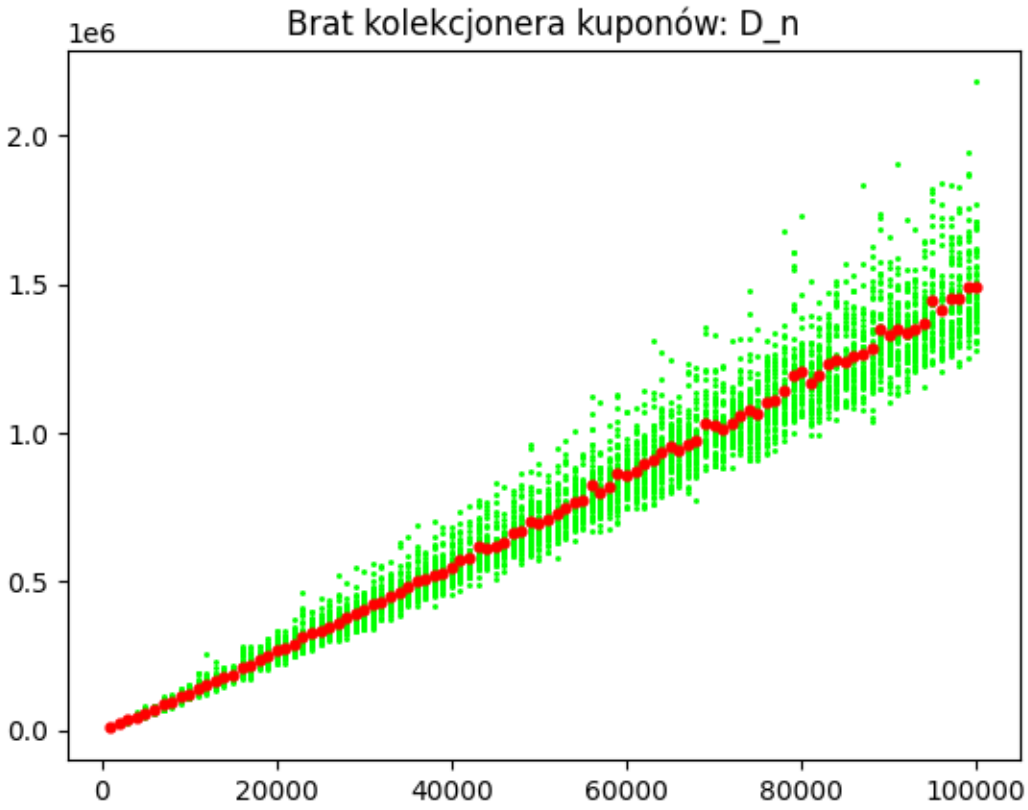
Program przeprowadził po $k = 50$ niezależnych prób dla każdego n z przedziału {1000, 2000, ..., 100000}, generując losowe liczby z przedziału {1, 2, ..., n} odpowiadające "wrzuceniu kuli do n-tej urny" dopóki w każdej urnie nie znajdowały się przynajmniej 2 kule. Poniżej znajdują się wykresy obrazujące uzyskane wyniki dla różnych wielkości (czerwone kropki oznaczają wyniki średnie):

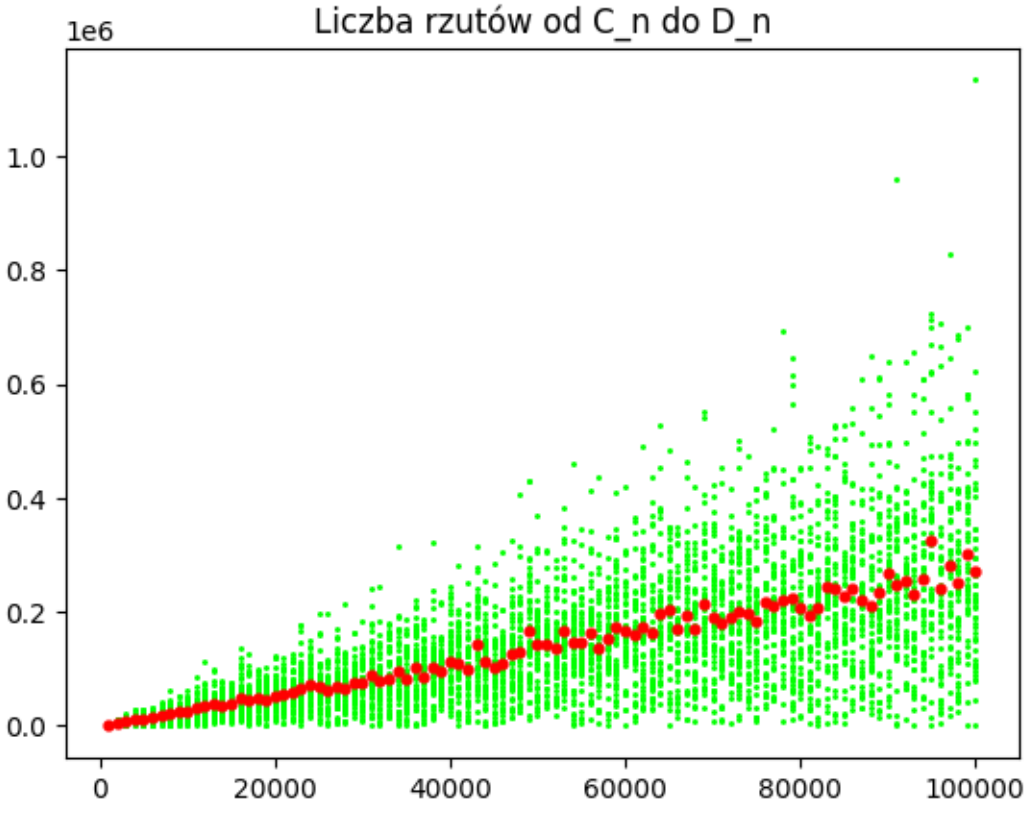





Maksymalna liczba kul w urnie po wrzuceniu n kul: L_n
10
9
8
7
6
5
4
0
20000
40000
60000
80000
100000


Problem kolekcjonera kuponów: C_n
1e6
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
20000
40000
60000
80000
100000


Brat kolekcjonera kuponów: D_n
1e6
2.0
1.5
1.0
0.5
0.0
0
20000
40000
60000
80000
100000

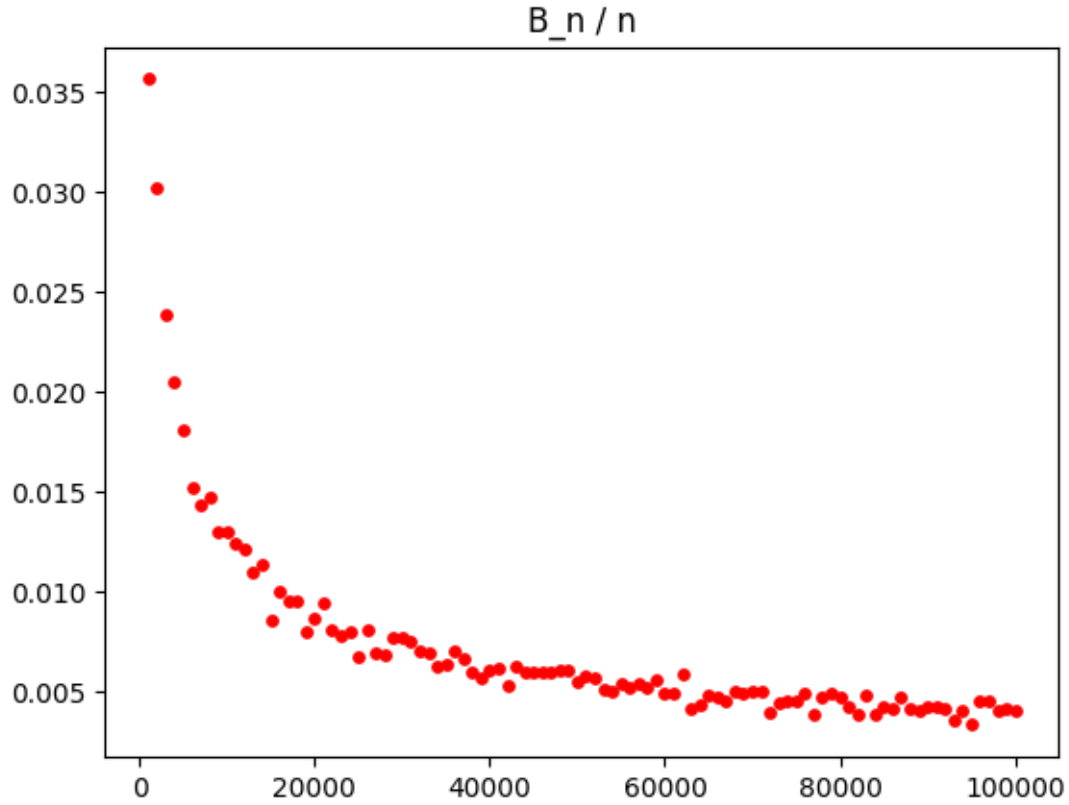

Liczba rzutów od C_n do D_n
1e6
1.0
0.8
0.6
0.4
0.2
0.0
0
20000
40000
60000
80000
100000

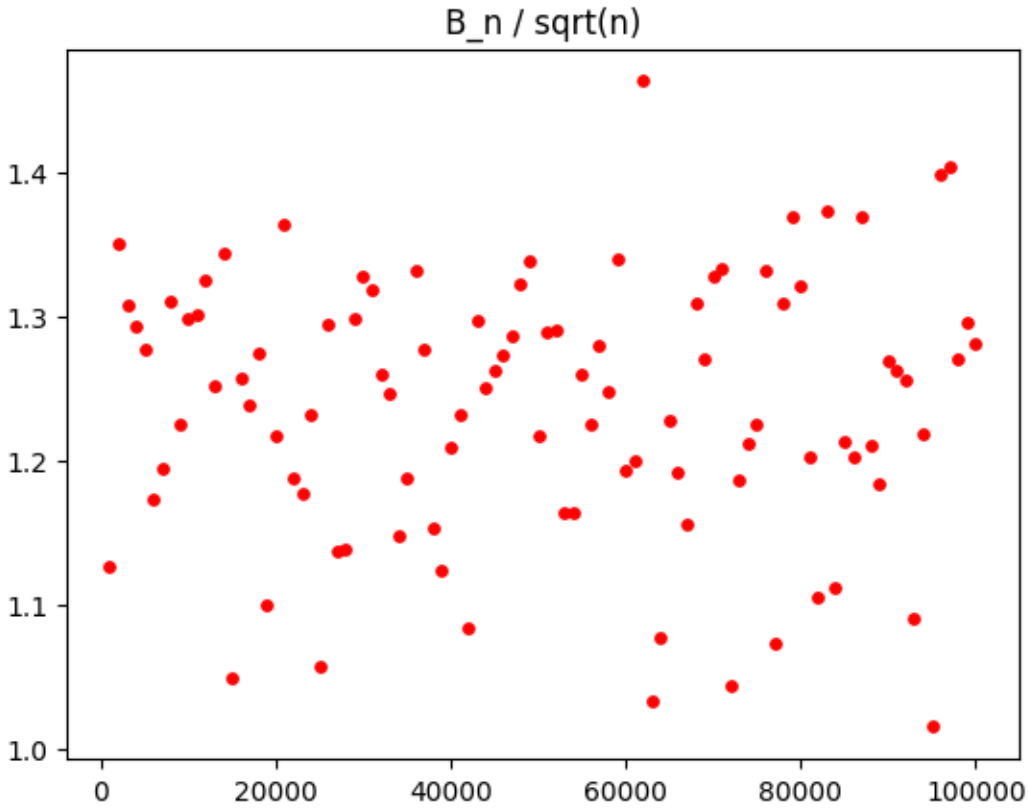
Mierzone wielkości to:

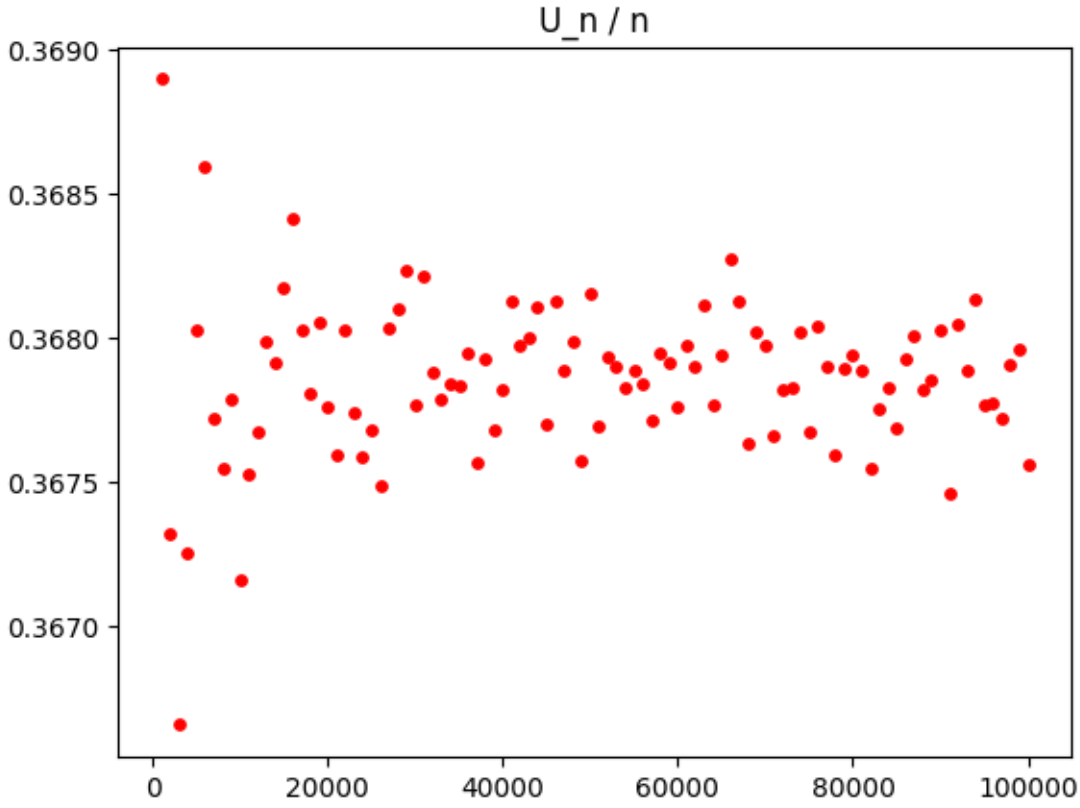
- B_n – moment pierwszej kolizji, równy *k* jeśli *k*-ta z wrzucanych kul jest pierwszą, która trafiła do niepustej urny (paradoks urodzinowy)
- U_n – liczba pustych urn po wrzuceniu *n* kul
- L_n – maksymalna liczba kul w urnie po wrzuceniu *n* kul
- C_n – minimalna liczba rzutów, po której w każdej z urn jest co najmniej jedna kula
- D_n – minimalna liczba rzutów, po której w każdej z urn są co najmniej dwie kule
- $D_n - C_n$ – liczba rzutów od momentu C_n potrzebna do tego, żeby w każdej urnie były co najmniej dwie kule

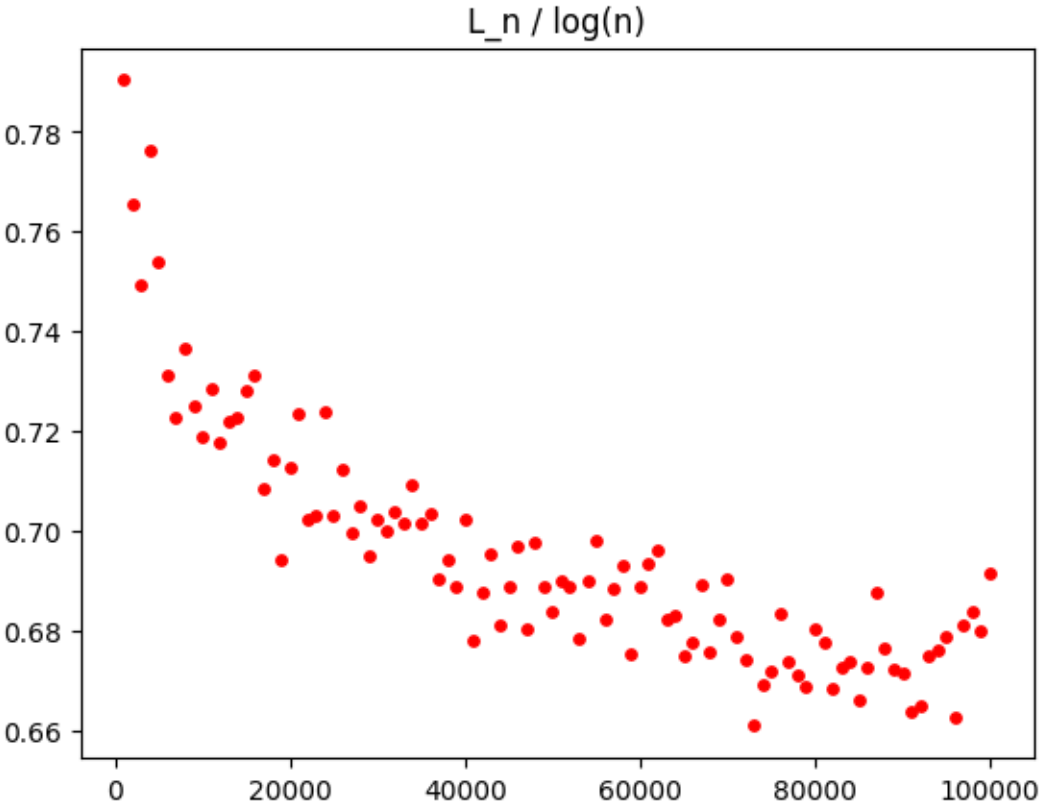
Jak widać z wykresów powyżej, wszystkie wartości średnie rosną wraz z wartością *n*, zgodnie z intuicją. Analizując wykres U_n dostrzegamy, że poszczególne punkty średnich wyznaczają z dużą dokładnością pewną prostą; ponadto, wartości dla przypadków poszczególnych mają tę samą własność. Dla wykresu L_n widać, że poszczególne wartości średnie są bardziej skoncentrowane dla większych wartości *n* – są one jednak słabo przybliżane przez wyniki poszczególnych eksperymentów (błąd względny pojedynczego wyniku może wynosić nawet około 25% nawet dla dużych *n*). Dla wartości B_n, C_n, D_n oraz $D_n - C_n$ obserwujemy, że precyzja estymacji zaczyna maleć wraz ze wzrostem *n* – średnia odległość poszczególnych średnich rośnie. To samo obserwujemy dla wyników poszczególnych doświadczeń – odchylenia od średnich wzrastają razem z wartością *n*.

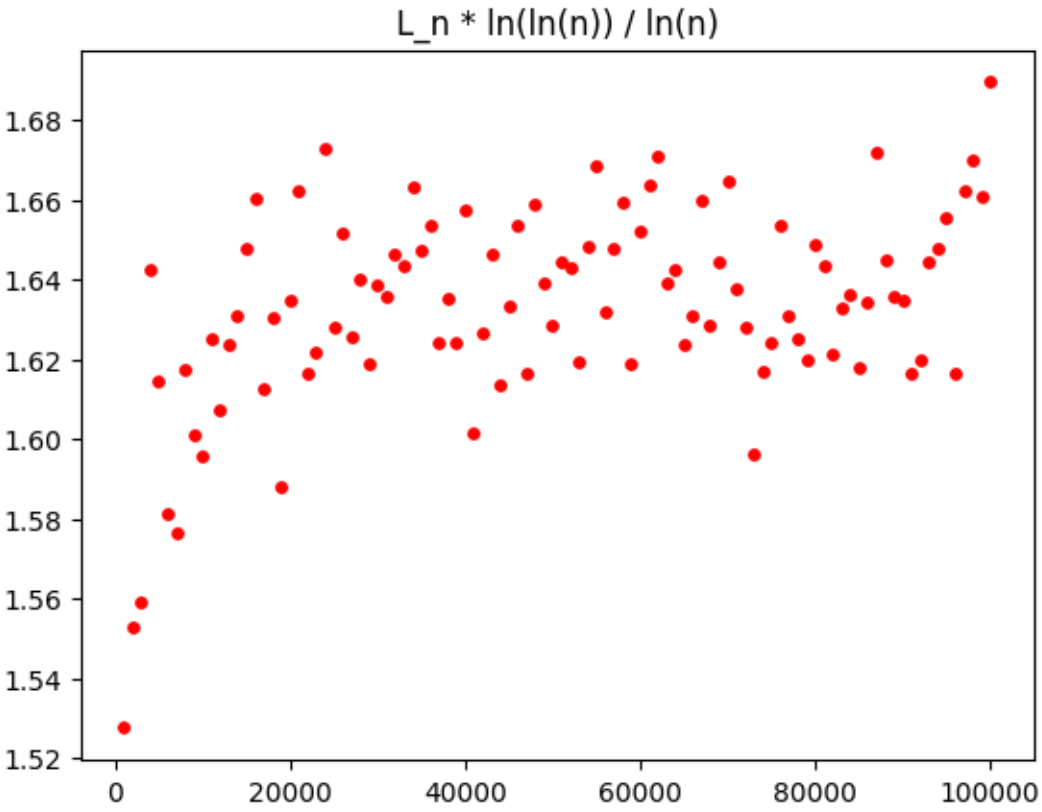
Poniższe wykresy ilustrują ilorazy średnich wartości poszczególnych wartości z wybranymi funkcjami zmiennej *n*:

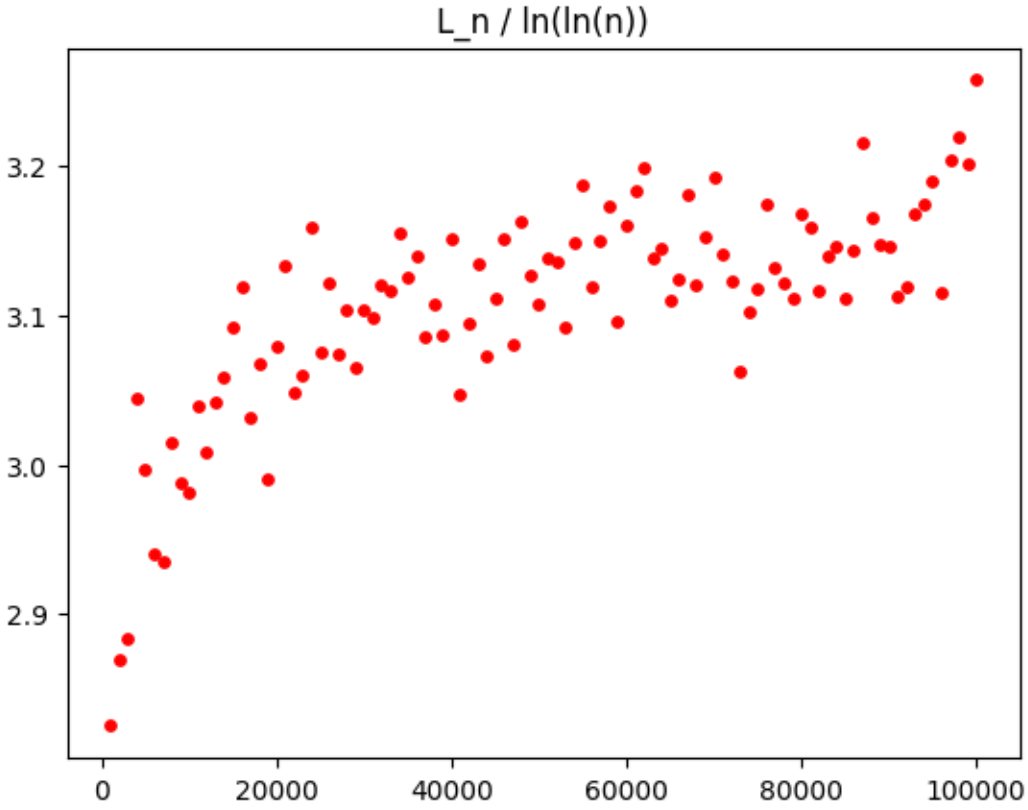


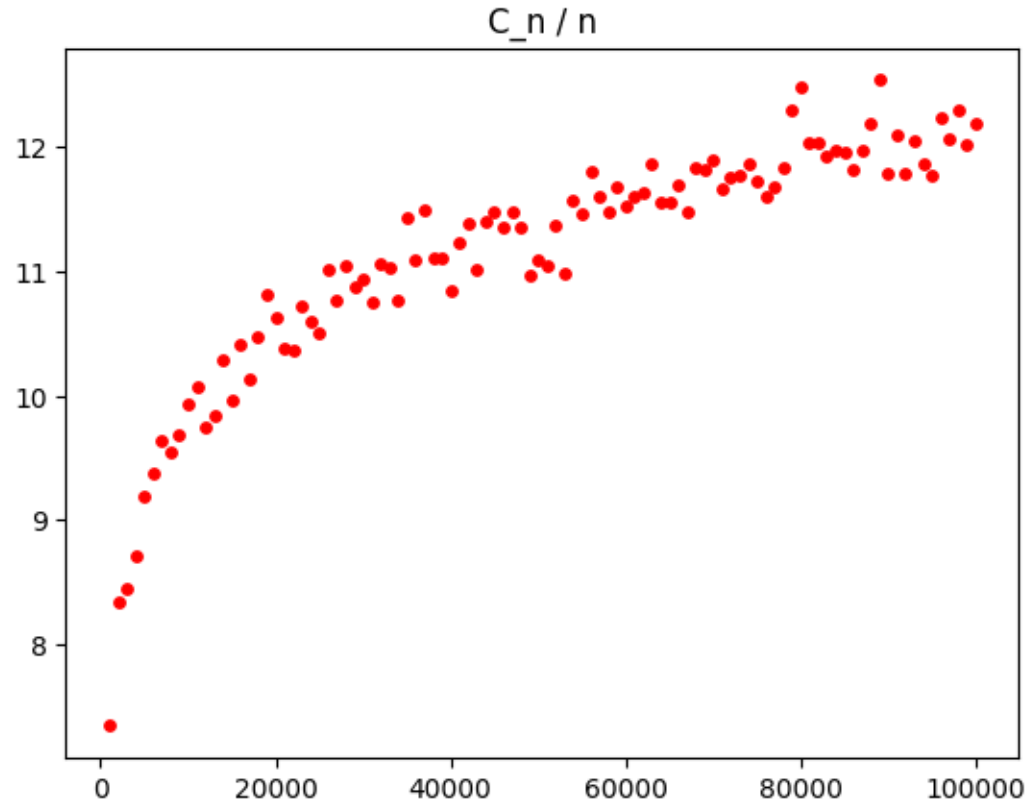

B_n / sqrt(n)
1.4
1.3
1.2
1.1
1.0
0
20000
40000
60000
80000
100000

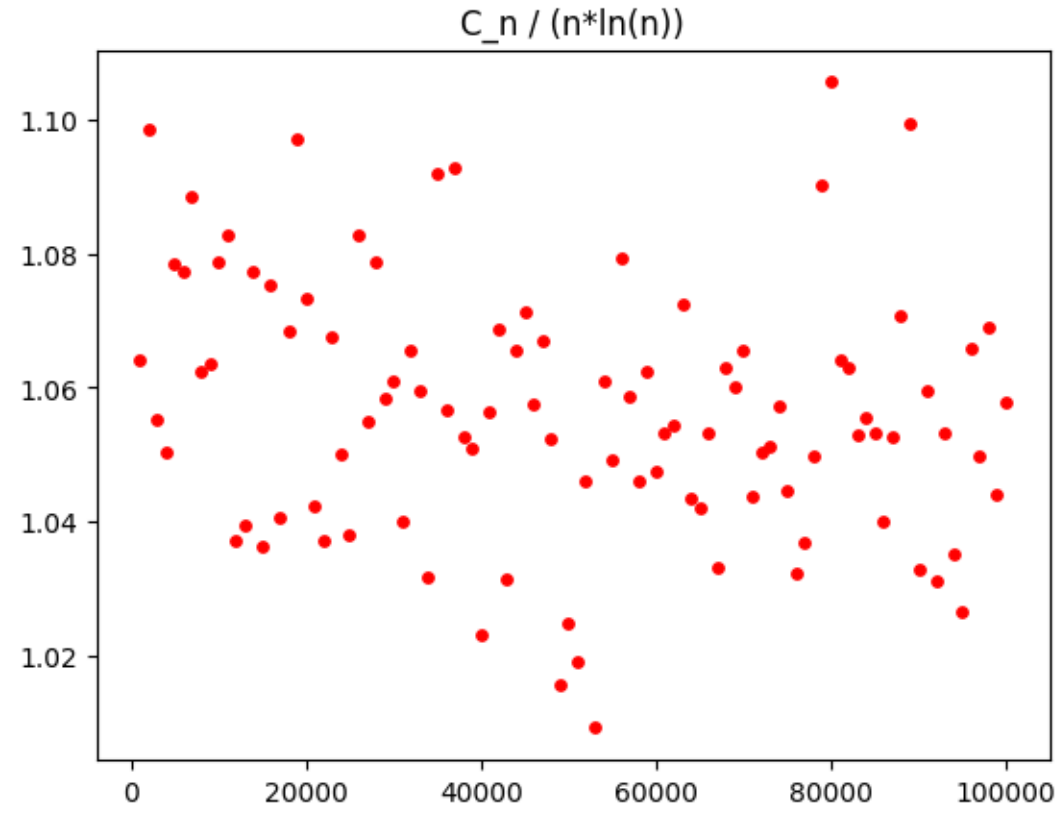

U_n / n
0.3690
0.3685
0.3680
0.3675
0.3670
0
20000
40000
60000
80000
100000

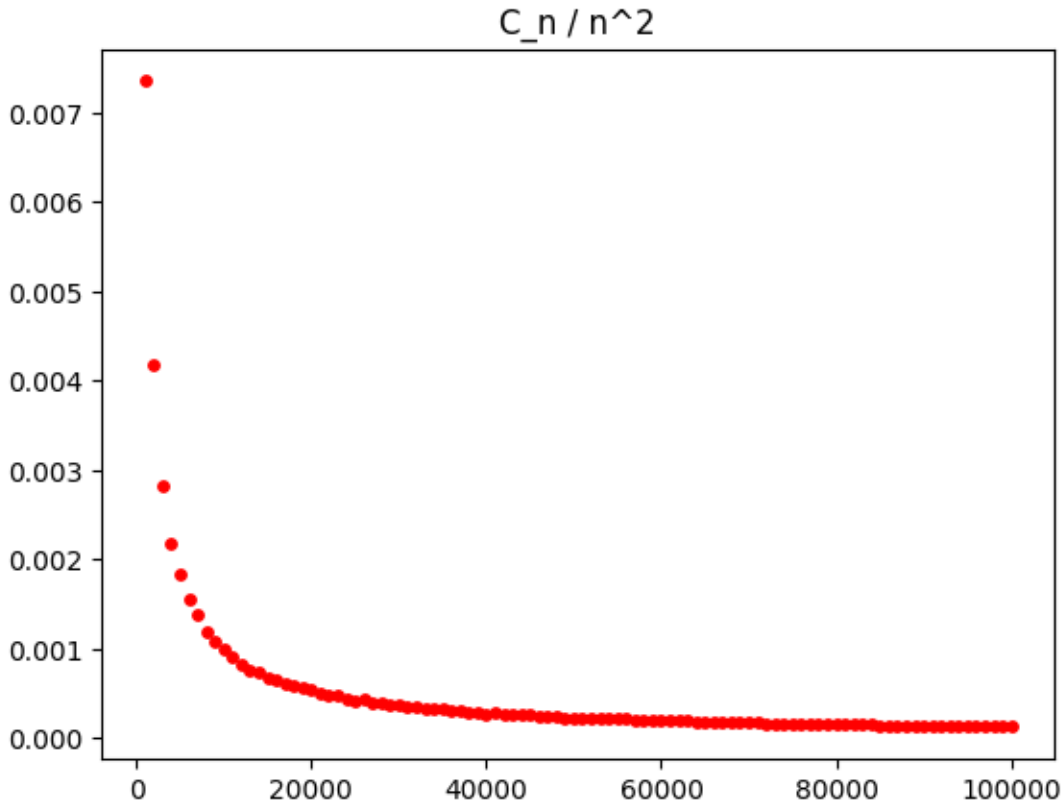

L_n / log(n)
0.78
0.76
0.74
0.72
0.70
0.68
0.66
0
20000
40000
60000
80000
100000

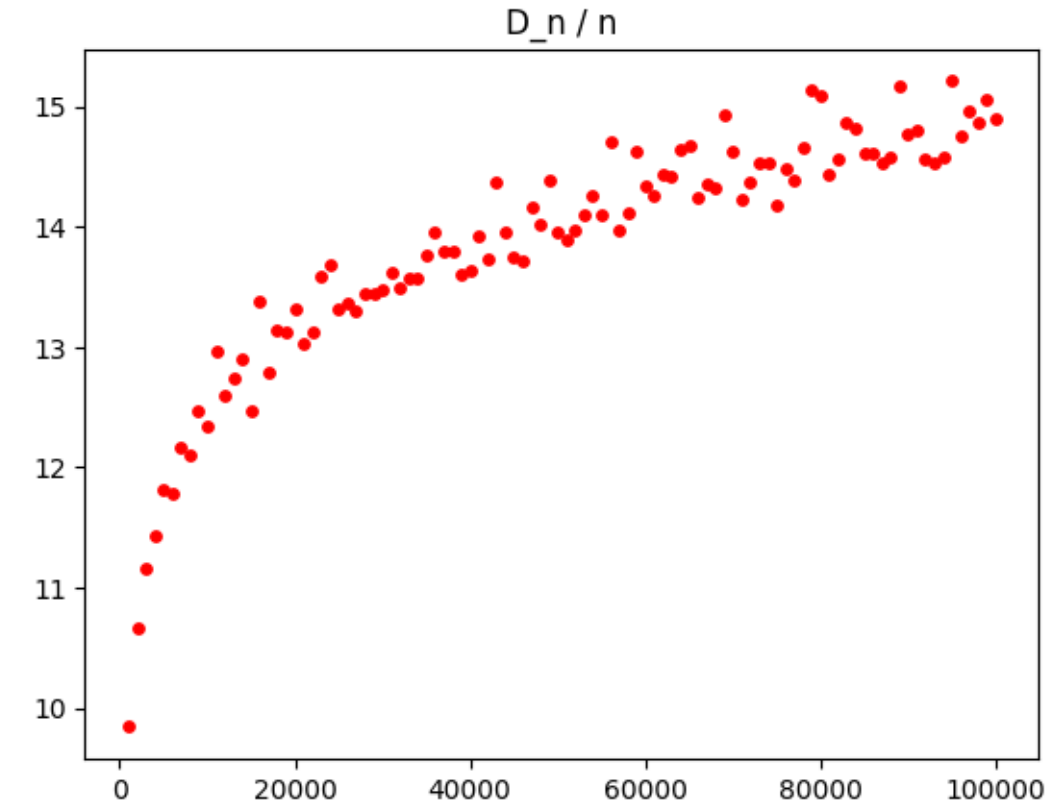

L_n * ln(ln(n)) / ln(n)
1.68
1.66
1.64
1.62
1.60
1.58
1.56
1.54
1.52
0
20000
40000
60000
80000
100000

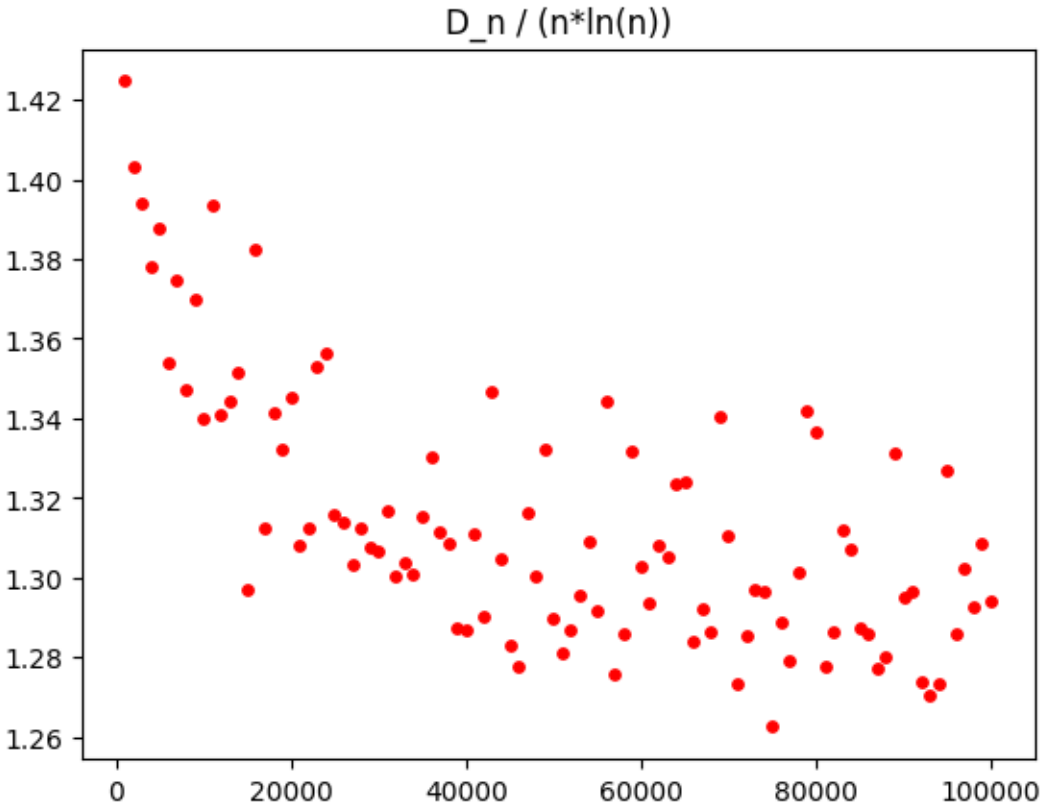

L_n / ln(ln(n))
3.2
3.1
3.0
2.9
0
20000
40000
60000
80000
100000

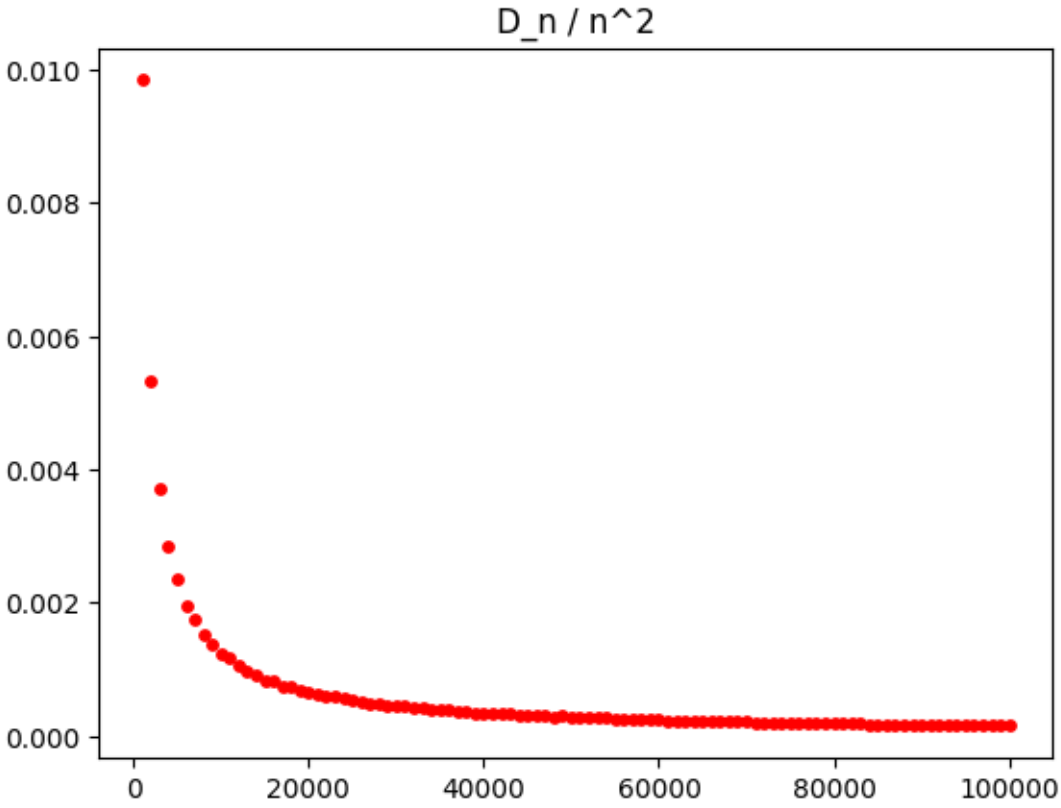

C_n / n
12
11
10
9
8
0
20000
40000
60000
80000
100000

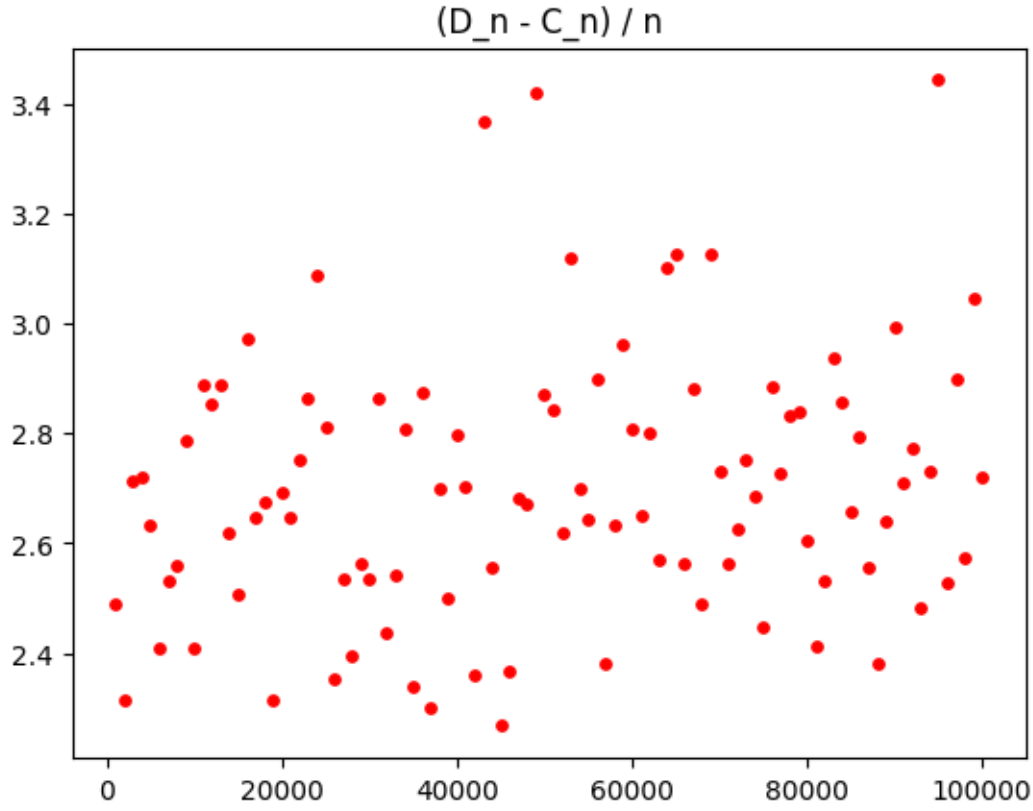

C_n / (n*ln(n))
1.10
1.08
1.06
1.04
1.02
0
20000
40000
60000
80000
100000

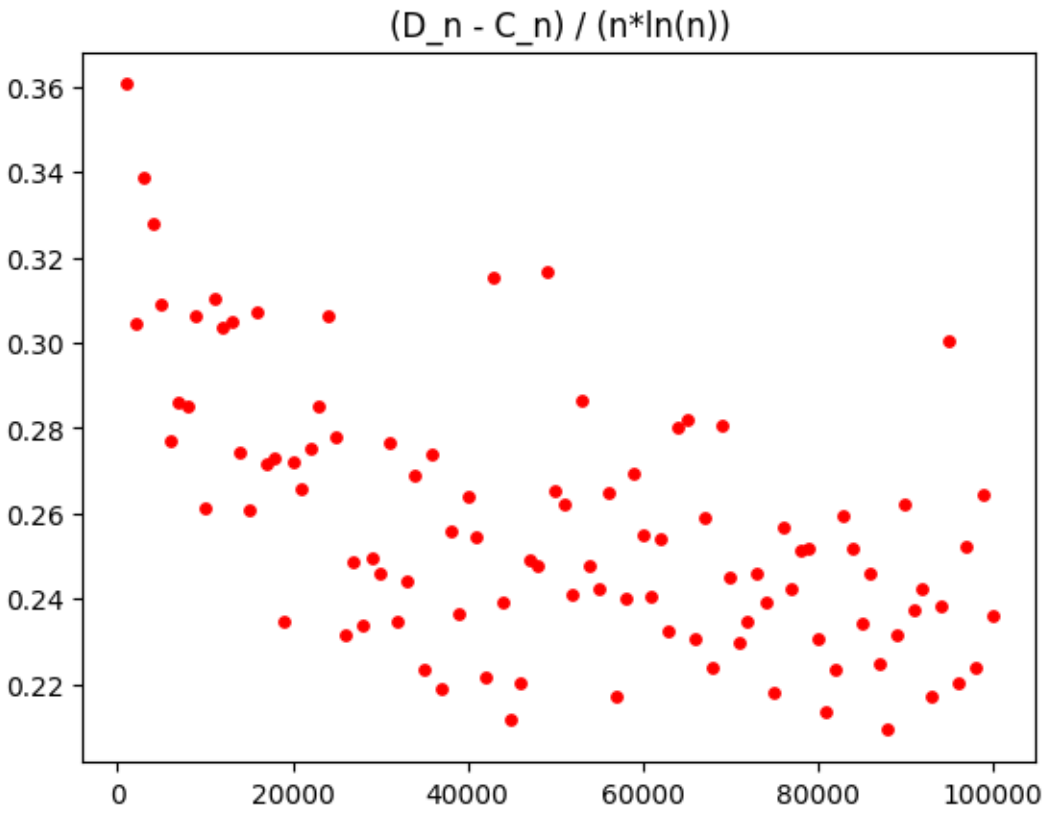

C_n / n^2
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
20000
40000
60000
80000
100000

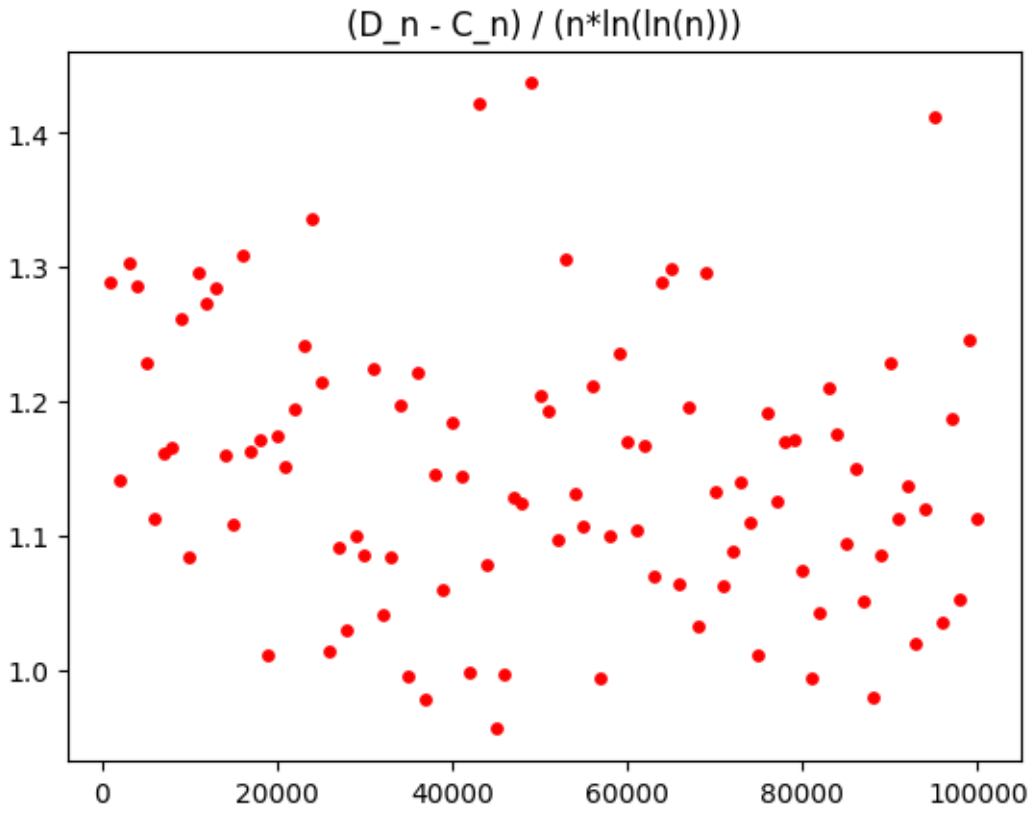

D_n / n
15
14
13
12
11
10
0
20000
40000
60000
80000
100000


D_n / (n*ln(n))
1.42
1.40
1.38
1.36
1.34
1.32
1.30
1.28
1.26
0
20000
40000
60000
80000
100000


D_n / n^2
0.010
0.008
0.006
0.004
0.002
0.000
0
20000
40000
60000
80000
100000


(D_n - C_n) / n
3.4
3.2
3.0
2.8
2.6
2.4
0
20000
40000
60000
80000
100000


(D_n - C_n) / (n*ln(n))
0.36
0.34
0.32
0.30
0.28
0.26
0.24
0.22
0
20000
40000
60000
80000
100000


(D_n - C_n) / (n*ln(ln(n)))
1.4
1.3
1.2
1.1
1.0
0
20000
40000
60000
80000
100000

Na podstawie powyższych wykresów możemy postawić hipotezy dotyczące asymptotyki wyznaczonych wartości:

- Bn = O(sqrt(n))
- Un = O(n)
- Ln = O(ln(n) / ln(ln(n)))
- Cn = O(n ln(n))
- Dn = O(n ln(n))
- Dn − Cn = O(n ln(ln(n)))

Nazwy "birthday paradox" oraz coupon collector's problem" można uzasadnić jak następuje:

- "birthday paradox" – wybierajmy jednostajnie losowo po jednym człowieku z populacji na świecie. Czekamy na moment pierwszej kolizji, czyli pierwszy moment kiedy dwóch ludzi ma tę samą datę urodzenia. wynik jest paradoksalny z powodu tego, że uzyskane wyniki są nieintuicyjnie małe.
- "coupon collector's problem" – pewien człowiek zbiera kupony. Szukamy momentu, po którym będzie miał kupony każdego typu. Uzyskane wartość jest odpowiedzią na postawiony problem.

Paradoks urodzinowy jest ważny ze względu na funkcje haszujące, ponieważ zależy nam na dostępie do shaszowanych danych ze złożonością O(1). Jeśli istnieją powtórzenia w wynikach haszy, to złożoność dostępu do danych rośnie. W kontekście kryptograficznych funkcji haszujących, musimy mieć pewność, że hasze haseł przechowywane w bazie danych są unikatowe – jeśli dwa różne hasła mają ten sam hasz, to dany program łamiący hasła musiałbym średnio spędzić 2x krócej łamiąc hasło użytkownika, lub można byłoby się zalogować jako dany użytkownik korzystając z niepoprawnego hasła!